

---

# NSSpeechSynthesizer Class Reference

[Cocoa](#) > [Accessibility](#)



2009-01-06



Apple Inc.  
© 2009 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Cocoa, iPod, Mac, Mac OS, and QuickTime are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY**

**DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

## **NSSpeechSynthesizer Class Reference 7**

---

Overview	7
Tasks	9
Creating Speech Synthesizers	9
Configuring Speech Synthesizers	9
Getting Speech Synthesizer Information	9
Getting Speech State	10
Synthesizing Speech	10
Getting Phonemes	10
Class Methods	11
attributesForVoice:	11
availableVoices	11
defaultVoice	11
isAnyApplicationSpeaking	12
Instance Methods	12
addSpeechDictionary:	12
continueSpeaking	12
delegate	13
initWithVoice:	13
isSpeaking	14
objectForProperty:error:	14
pauseSpeakingAtBoundary:	14
phonemesFromText:	15
rate	15
setDelegate:	16
setObject:forProperty:error:	16
setRate:	16
setUsesFeedbackWindow:	17
setVoice:	17
setVolume:	18
startSpeakingString:	18
startSpeakingString:toURL:	19
stopSpeaking	19
stopSpeakingAtBoundary:	20
usesFeedbackWindow	20
voice	21
volume	21
Delegate Methods	21
speechSynthesizer:didEncounterErrorAtIndex:ofString:message:	21
speechSynthesizer:didEncounterSyncMessage:	22
speechSynthesizer:didFinishSpeaking:	22

- speechSynthesizer:willSpeakPhoneme: 23
- speechSynthesizer:willSpeakWord:ofString: 23
- Constants 24
  - Voice Attributes 24
  - Voice Genders 26
  - Speech Synthesizer Property Keys 26
  - Input Modes 27
  - Speech Type 27
  - Speech Status 27
  - Synthesis Error 28
  - Synthesizer Information 28
  - Speech Command Delimiter 28
  - Dictionary Properties 28
  - NSSpeechBoundary 28

---

**Document Revision History 31**

---

**Index 33**

---

# Figures, Tables, and Listings

## **NSSpeechSynthesizer Class Reference** 7

---

Figure 1	Speech feedback window	8
Table 1	NSSpeechSynthesizer instance attributes	8
Listing 1	Identifiers of the Mac OS X system voices	25



# NSSpeechSynthesizer Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Availability</b>	Available in Mac OS X v10.3 and later.
<b>Companion guide</b>	Speech
<b>Declared in</b>	NSSpeechSynthesizer.h
<b>Related sample code</b>	SayIt

## Overview

The `NSSpeechSynthesizer` class is the Cocoa interface to Speech Synthesis on Mac OS X. Instances of this class are called **speech synthesizers**.

Speech Synthesis, also called text-to-speech (TTS), parses text and converts it into audible speech. It offers a concurrent feedback mode that can be used in concert with or in place of traditional visual and aural notifications. For example, your application can use an `NSSpeechSynthesizer` object to “pronounce” the text of important alert dialogs. Synthesized speech has several advantages. It can provide urgent information to users without forcing them to shift attention from their current task. And because speech doesn’t rely on visual elements for meaning, it is a crucial technology for users with vision or attention disabilities.

In addition, synthesized speech can help save system resources. Because sound samples can take up large amounts of room on disk, using text in place of sampled sound is extremely efficient, and so a multimedia application might use an `NSSpeechSynthesizer` object to provide a narration of a QuickTime movie instead of including sampled-sound data on a movie track.

When you create an `NSSpeechSynthesizer` instance using the default initializer (`init`), the class uses the **default voice** selected in System Preferences > Speech. Alternatively, you can select a specific voice for an `NSSpeechSynthesizer` instance by initializing it with `initWithVoice:` (page 13). To begin synthesis, send either `startSpeakingString:` (page 18) or `startSpeakingString:toURL:` (page 19) to the instance. The former generates speech through the system’s default sound output device; the latter saves the generated speech to a file. If you wish to be notified when the current speech concludes, set a delegate object using `setDelegate:` (page 16) and implement the delegate method `speechSynthesizer:didFinishSpeaking:` (page 22).

Speech Synthesis is just one of the Mac OS X speech technologies. The Speech Recognizer technology allows applications to “listen to” text spoken in U.S. English; the `NSSpeechRecognizer` class is the Cocoa interface to this technology. Both technologies provide benefits for all users, and are particularly useful to those users who have difficulties seeing the screen or using the mouse and keyboard.

Table 1 lists the attributes of instances of this class. The list following the table describes each attribute.

**Table 1** NSSpeechSynthesizer instance attributes

Instance attribute	Default value	Accessors
<code>UsesFeedbackWindow</code>	YES	<code>usesFeedbackWindow</code> (page 20), <code>setUsesFeedbackWindow:</code> (page 17)
<code>CurrentVoice</code>	Identifier of the default voice	<code>voice</code> (page 21), <code>setVoice:</code> (page 17)

- `UsesFeedbackWindow`

The speech feedback window (Figure 1) displays the text recognized from the user’s speech and the text from which an `NSSpeechSynthesizer` object synthesizes speech. Using the feedback window makes spoken exchange more natural and helps the user understand the synthesized speech.

**Figure 1** Speech feedback window



For example, your application may use an `NSSpeechRecognizer` object to listen for the command “Play some music.” When it recognizes this command, your application might then respond by speaking “Which artist?” using a speech synthesizer.

When `UsesFeedbackWindow` is YES, the speech synthesizer uses the feedback window if its visible, which the user specifies in System Preferences > Speech.

- `CurrentVoice`

The identifier of the voice the receiver uses to speak synthesized text.



## Tasks

### Creating Speech Synthesizers

- [initWithVoice:](#) (page 13)  
Initializes the receiver with a voice.

### Configuring Speech Synthesizers

- [usesFeedbackWindow](#) (page 20)  
Indicates whether the receiver uses the speech feedback window.
- [setUsesFeedbackWindow:](#) (page 17)  
Specifies whether the receiver uses the speech feedback window.
- [voice](#) (page 21)  
Returns the identifier of the receiver's current voice.
- [setVoice:](#) (page 17)  
Sets the receiver's current voice.
- [rate](#) (page 15)  
Provides the receiver's speaking rate.
- [setRate:](#) (page 16)  
Specifies the receivers speaking rate.
- [volume](#) (page 21)  
Provides the receiver's speaking volume.
- [setVolume:](#) (page 18)  
Specifies the receiver's speaking volume.
- [addSpeechDictionary:](#) (page 12)  
Registers the given speech dictionary with the receiver.
- [objectForProperty:error:](#) (page 14)  
Provides the value of a receiver's property.
- [setObject:forProperty:error:](#) (page 16)  
Specifies the value of a receiver's property.
- [delegate](#) (page 13)  
Returns the receiver's delegate.
- [setDelegate:](#) (page 16)  
Sets the receiver's delegate.

### Getting Speech Synthesizer Information

- + [availableVoices](#) (page 11)  
Provides the identifiers of the voices available on the system.
- + [attributesForVoice:](#) (page 11)  
Provides the attribute dictionary of a voice.

- + [defaultVoice](#) (page 11)  
Provides the identifier of the default voice.

## Getting Speech State

- + [isAnyApplicationSpeaking](#) (page 12)  
Indicates whether any other application is currently speaking through the sound output device.

## Synthesizing Speech

- [isSpeaking](#) (page 14)  
Indicates whether the receiver is currently generating synthesized speech.
- [startSpeakingString:](#) (page 18)  
Begins speaking synthesized text through the system's default sound output device.
- [startSpeakingString:toURL:](#) (page 19)  
Begins synthesizing text into a sound (AIFF) file.
- [speechSynthesizer:willSpeakWord:ofString:](#) (page 23) *delegate method*  
Sent just before a synthesized word is spoken through the sound output device.
- [speechSynthesizer:willSpeakPhoneme:](#) (page 23) *delegate method*  
Sent just before a synthesized phoneme is spoken through the sound output device.
- [pauseSpeakingAtBoundary:](#) (page 14)  
Pauses synthesis in progress at a given boundary.
- [continueSpeaking](#) (page 12)  
Resumes synthesis.
- [stopSpeaking](#) (page 19)  
Stops synthesis in progress.
- [stopSpeakingAtBoundary:](#) (page 20)  
Stops synthesis in progress at a given boundary.
- [speechSynthesizer:didEncounterErrorAtIndex:ofString:message:](#) (page 21) *delegate method*  
Sent to the delegate when a speech synthesizer encounters an error in text being synthesized.
- [speechSynthesizer:didEncounterSyncMessage:](#) (page 22) *delegate method*  
Sent to the delegate when a speech synthesizer encounters a synchronization error.
- [speechSynthesizer:didFinishSpeaking:](#) (page 22) *delegate method*  
Sent when an [NSSpeechSynthesizer](#) (page 7) object finishes speaking through the sound output device.

## Getting Phonemes

- [phonemesFromText:](#) (page 15)  
Provides the phoneme symbols generated by the given text.

## Class Methods

### attributesForVoice:

Provides the attribute dictionary of a voice.

```
+ (NSDictionary *)attributesForVoice:(NSString *)voiceIdentifier
```

#### Parameters

*voiceIdentifier*

Identifier of the voice whose attributes you want to obtain.

#### Return Value

Attribute dictionary of the voice identified by *voiceIdentifier*.

#### Discussion

The keys and values of voice attribute dictionaries are described in “[Constants](#)” (page 24).

#### Availability

Available in Mac OS X v10.3 and later.

#### Declared In

NSSpeechSynthesizer.h

### availableVoices

Provides the identifiers of the voices available on the system.

```
+ (NSArray *)availableVoices
```

#### Return Value

Array of strings representing the identifiers of each voice available on the system.

#### Availability

Available in Mac OS X v10.3 and later.

#### See Also

+ [attributesForVoice:](#) (page 11)

- [setVoice:](#) (page 17)

#### Declared In

NSSpeechSynthesizer.h

### defaultVoice

Provides the identifier of the default voice.

```
+ (NSString *)defaultVoice
```

#### Return Value

Identifier of the default voice.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSSpeechSynthesizer.h

**isAnyApplicationSpeaking**

Indicates whether any other application is currently speaking through the sound output device.

+ (BOOL)isAnyApplicationSpeaking

**Return Value**

YES when another application is producing speech through the sound output device, NO otherwise.

**Discussion**

You usually invoke this method to prevent your application from speaking over speech being generated by another application or system component.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSSpeechSynthesizer.h

## Instance Methods

**addSpeechDictionary:**

Registers the given speech dictionary with the receiver.

- (void)addSpeechDictionary:(NSDictionary \*)*speechDictionary*

**Parameters**

*speechDictionary*

Speech dictionary to add to the receiver's dictionaries.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

NSSpeechSynthesizer.h

**continueSpeaking**

Resumes synthesis.

- (void)continueSpeaking

**Discussion**

If synthesis is paused, it's resumed.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [pauseSpeakingAtBoundary:](#) (page 14)

**Declared In**

NSSpeechSynthesizer.h

**delegate**

Returns the receiver's delegate.

- (id)delegate

**Return Value**

The receiver's delegate.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [setDelegate:](#) (page 16)

**Declared In**

NSSpeechSynthesizer.h

**initWithVoice:**

Initializes the receiver with a voice.

- (id)initWithVoice:(NSString \*)*voiceIdentifier*

**Parameters**

*voiceIdentifier*

Identifier of the voice to set as the current voice. When `nil`, the default voice is used. Passing in a specific voice means the initial speaking rate is determined by the synthesizer's default speaking rate; passing `nil` means the speaking rate is automatically set to the rate the user specifies in Speech preferences.

**Return Value**

Initialized speech synthesizer or `nil` when the voice identified by *voiceIdentifier* is not available or when there's an allocation error.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

+ [availableVoices](#) (page 11)

**Declared In**

NSSpeechSynthesizer.h

**isSpeaking**

Indicates whether the receiver is currently generating synthesized speech.

- (BOOL)isSpeaking

**Return Value**

YES when the receiver is generating synthesized speech, NO otherwise.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [startSpeakingString:](#) (page 18)
- [startSpeakingString:toURL:](#) (page 19)
- [stopSpeaking](#) (page 19)

**Declared In**

NSSpeechSynthesizer.h

**objectForProperty:error:**

Provides the value of a receiver's property.

- (id)objectForProperty:(NSString \*)*speechProperty* error:(NSError \*\*)*out\_error***Parameters***speechProperty*

Property to get.

*out\_error*On output, error that ocured while obtaining the value of *speechProperty*.**Return Value**The value of *speechProperty*.**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [setObject:forProperty:error:](#) (page 16)

**Declared In**

NSSpeechSynthesizer.h

**pauseSpeakingAtBoundary:**

Pauses synthesis in progress at a given boundary.

- (void)pauseSpeakingAtBoundary:(NSSpeechBoundary) *boundary*

**Parameters**

*boundary*

Boundary at which to pause speech..

**Discussion**

If the receiver is currently generating speech, synthesis is paused at boundary.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [continueSpeaking](#) (page 12)

**Declared In**

NSSpeechSynthesizer.h

## phonemesFromText:

Provides the phoneme symbols generated by the given text.

- (NSString \*)phonemesFromText:(NSString \*) *text*

**Parameters**

*text*

Text from which to generate phonemes.

**Return Value**

Phonemes generated from text.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

NSSpeechSynthesizer.h

## rate

Provides the receiver's speaking rate.

- (float)rate

**Return Value**

Speaking rate (words per minute).

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [setRate:](#) (page 16)

**Declared In**

NSSpeechSynthesizer.h

**setDelegate:**

Sets the receiver's delegate.

```
- (void)setDelegate:(id)delegate
```

**Parameters**

*delegate*

Object to be the receiver's delegate.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [delegate](#) (page 13)

**Declared In**

NSSpeechSynthesizer.h

**setObject:forProperty:error:**

Specifies the value of a receiver's property.

```
- (BOOL)setObject:(id)object forProperty:(NSString *) speechProperty error:(NSError **)out_error
```

**Parameters**

*speechProperty*

Property to set.

*out\_error*

On output, error that occurred while setting *speechProperty*.

**Return Value**

YES when the *speechProperty* was set. NO when there was an error, specified in *out\_error*.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [objectForProperty:error:](#) (page 14)

**Declared In**

NSSpeechSynthesizer.h

**setRate:**

Specifies the receivers speaking rate.

```
- (void)setRate:(float)rate
```

**Parameters**

*rate*

Words to speak per minute.



**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [rate](#) (page 15)

**Declared In**

NSSpeechSynthesizer.h

**setUsesFeedbackWindow:**

Specifies whether the receiver uses the speech feedback window.

- (void)setUsesFeedbackWindow:(BOOL)useFeedbackWindow

**Parameters**

*useFeedbackWindow*

YES to make the receiver use the speech feedback window if it's visible when the user begins speaking.

NO not to use the feedback window.

**Discussion**

See the class description for details on the `UsesFeedbackWindow` attribute.

**Important:** The delegate does not receive [speechSynthesizer:willSpeakWord:ofString:](#) (page 23) and [speechSynthesizer:willSpeakPhoneme:](#) (page 23) messages when speaking occurs through the feedback window.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [usesFeedbackWindow](#) (page 20)

**Declared In**

NSSpeechSynthesizer.h

**setVoice:**

Sets the receiver's current voice.

- (BOOL)setVoice:(NSString \*)voiceIdentifier

**Parameters**

*voiceIdentifier*

Identifier of the voice to set at the receiver's current voice. When `nil`, the receiver sets the default voice as its current voice.

**Return Value**

YES when the receiver is not currently synthesizing speech and the current voice is set successfully, NO otherwise.

**Discussion**

See the class description for details on the `CurrentVoice` attribute.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [voice](#) (page 21)

+ [defaultVoice](#) (page 11)

**Declared In**

`NSSpeechSynthesizer.h`

**setVolume:**

Specifies the receiver's speaking volume.

```
- (void)setVolume:(float)volume
```

**Parameters**

*volume*

Sound level to use for speech.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [volume](#) (page 21)

**Declared In**

`NSSpeechSynthesizer.h`

**startSpeakingString:**

Begins speaking synthesized text through the system's default sound output device.

```
- (BOOL)startSpeakingString:(NSString *)text
```

**Parameters**

*text*

Text to synthesize. When `nil` or empty, no synthesis occurs.

**Return Value**

YES when synthesis starts successfully, NO otherwise.

**Discussion**

If the receiver is currently speaking synthesized speech when `startSpeakingString:` is called, that process is stopped before *text* is spoken.

When synthesis of *text* finishes normally or is stopped, the message [speechSynthesizer:didFinishSpeaking:](#) (page 22) is sent to the delegate.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [isSpeaking](#) (page 14)
- [startSpeakingString:toURL:](#) (page 19)
- [stopSpeaking](#) (page 19)

**Declared In**

NSSpeechSynthesizer.h

**startSpeakingString:toURL:**

Begins synthesizing text into a sound (AIFF) file.

```
- (BOOL)startSpeakingString:(NSString *)text toURL:(NSURL *)url
```

**Parameters***text*Text to synthesize. When *nil* or empty, no synthesis is started.*url*

Filesystem location of the output sound file.

**Return Value**

YES when synthesis starts successfully, NO otherwise.

**Discussion**

When synthesis of *text* finishes normally or is stopped, the message [speechSynthesizer:didFinishSpeaking:](#) (page 22) is sent to the delegate.

One example of how you might use this method is in an email program that automatically converts new messages into sound files that can be stored on an iPod for later listening.

**Note:** The delegate does not receive [speechSynthesizer:willSpeakWord:ofString:](#) (page 23) and [speechSynthesizer:willSpeakPhoneme:](#) (page 23) messages when text is being synthesized to a file.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [isSpeaking](#) (page 14)
- [startSpeakingString:](#) (page 18)
- [stopSpeaking](#) (page 19)

**Declared In**

NSSpeechSynthesizer.h

**stopSpeaking**

Stops synthesis in progress.

```
- (void)stopSpeaking
```

**Discussion**

If the receiver is currently generating speech, synthesis is halted, and the message [speechSynthesizer:didFinishSpeaking:](#) (page 22) is sent to the delegate.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [isSpeaking](#) (page 14)
- [startSpeakingString:](#) (page 18)
- [startSpeakingString:toURL:](#) (page 19)

**Declared In**

NSSpeechSynthesizer.h

**stopSpeakingAtBoundary:**

Stops synthesis in progress at a given boundary.

- (void)stopSpeakingAtBoundary:(NSSpeechBoundary) *boundary*

**Parameters**

*boundary*

Boundary at which to stop speech.

**Discussion**

If the receiver is currently generating speech, synthesis is halted at boundary, and the message [speechSynthesizer:didFinishSpeaking:](#) (page 22) is sent to the delegate.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

NSSpeechSynthesizer.h

**usesFeedbackWindow**

Indicates whether the receiver uses the speech feedback window.

- (BOOL)usesFeedbackWindow

**Return Value**

YES when the receiver uses the speech feedback window, NO otherwise.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [setUsesFeedbackWindow:](#) (page 17)

**Declared In**

NSSpeechSynthesizer.h

## voice

Returns the identifier of the receiver's current voice.

- (NSString \*)voice

### Return Value

Identifier of the receiver's current voice.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [setVoice:](#) (page 17)

### Declared In

NSSpeechSynthesizer.h

## volume

Provides the receiver's speaking volume.

- (float)volume

### Return Value

Speaking volume: From 0.0 (minimum) to 1.0 (maximum).

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [setVolume:](#) (page 18)

### Declared In

NSSpeechSynthesizer.h

## Delegate Methods

### speechSynthesizer:didEncounterErrorAtIndex:ofString:message:

Sent to the delegate when a speech synthesizer encounters an error in text being synthesized.

```
- (void)speechSynthesizer:(NSSpeechSynthesizer *)sender
    didEncounterErrorAtIndex:(NSUInteger)characterIndex ofString:(NSString *)text
    message:(NSString *)errorMessage
```

#### Parameters

*sender*

Speech synthesizer informing its delegate of an error.

*characterIndex*

Location in text where the reciver encountered the error.

*text*

Text the receiver was synthesizing when the error occurred.

*errorMessage*

Error message.

#### Availability

Available in Mac OS X v10.5 and later.

#### Declared In

NSSpeechSynthesizer.h

## speechSynthesizer:didEncounterSyncMessage:

Sent to the delegate when a speech synthesizer encounters a synchronization error.

```
- (void)speechSynthesizer:(NSSpeechSynthesizer *)sender
  didEncounterSyncMessage:(NSString *)errorMessage
```

#### Parameters

*sender*

Speech synthesizer informing its delegate of an error.

*errorMessage*

Error message.

#### Availability

Available in Mac OS X v10.5 and later.

#### Declared In

NSSpeechSynthesizer.h

## speechSynthesizer:didFinishSpeaking:

Sent when an [NSSpeechSynthesizer](#) (page 7) object finishes speaking through the sound output device.

```
- (void)speechSynthesizer:(NSSpeechSynthesizer *)sender
  didFinishSpeaking:(BOOL)success
```

#### Parameters

*sender*

An [NSSpeechSynthesizer](#) (page 7) object that has stopped speaking into the sound output device.

*success*

YES when speaking completed normally, NO if speaking is stopped prematurely for any reason.

#### Availability

Available in Mac OS X v10.3 and later.

#### See Also

- [startSpeakingString:](#) (page 18)
- [stopSpeaking](#) (page 19)

**Declared In**

NSSpeechSynthesizer.h

**speechSynthesizer:willSpeakPhoneme:**

Sent just before a synthesized phoneme is spoken through the sound output device.

```
(void)speechSynthesizer:(NSSpeechSynthesizer *)sender
willSpeakPhoneme:(short)phonemeOpcode
```

**Parameters***sender*An [NSSpeechSynthesizer](#) (page 7) object that's synthesizing text into speech.*phonemeOpcode*Phoneme that *sender* is about to speak into the sound output device.**Discussion**

One use of this method might be to animate a mouth on screen to match the generated speech.

**Important:** The delegate is not sent this message when the `NSSpeechSynthesizer` object is synthesizing speech to a file ([startSpeakingString:toURL:](#) (page 19)).

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**- [startSpeakingString:](#) (page 18)**Declared In**

NSSpeechSynthesizer.h

**speechSynthesizer:willSpeakWord:ofString:**

Sent just before a synthesized word is spoken through the sound output device.

```
(void)speechSynthesizer:(NSSpeechSynthesizer *)sender
willSpeakWord:(NSRange)wordToSpeak ofString:(NSString *)text
```

**Parameters***sender*An [NSSpeechSynthesizer](#) (page 7) object that's synthesizing text into speech.*wordToSpeak*Word that *sender* is about to speak into the sound output device.*text*Text that is being synthesized by *sender*.**Discussion**

One use of this method might be to visually highlight the word being spoken.

**Important:** The delegate is not sent this message when the `NSSpeechSynthesizer` object is synthesizing speech to a file (`startSpeakingString:toURL:` (page 19)).

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- `startSpeakingString:` (page 18)

**Declared In**

`NSSpeechSynthesizer.h`

## Constants

### Voice Attributes

The following voice attributes are defined by the `NSSpeechSynthesizer` class for the dictionary returned by `attributesForVoice:` (page 11) (the values are all `NSString` objects).

```
NSString *NSVoiceIdentifier;
NSString *NSVoiceName;
NSString *NSVoiceAge;
NSString *NSVoiceGender;
NSString *NSVoiceDemoText;
NSString *NSVoiceLanguage;
NSString *NSVoiceLocaleIdentifier;
NSString *NSVoiceSupportedCharacters;
NSString *NSVoiceIndividuallySpokenCharacters;
```

**Constants**

`NSVoiceIdentifier`

A unique string identifying the voice. The identifiers of the system voices are listed in [Listing 1](#) (page 25).

Available in Mac OS X v10.3 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSVoiceName`

The name of the voice suitable for display.

Available in Mac OS X v10.3 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSVoiceAge`

The perceived age (in years) of the voice.

Available in Mac OS X v10.3 and later.

Declared in `NSSpeechSynthesizer.h`.



NSVoiceGender

The perceived gender of the voice. May be either NSVoiceGenderNeuter, NSVoiceGenderFemale, or NSVoiceGenderMale.

Available in Mac OS X v10.3 and later.

Declared in NSSpeechSynthesizer.h.

NSVoiceDemoText

A demonstration string to speak.

Available in Mac OS X v10.3 and later.

Declared in NSSpeechSynthesizer.h.

NSVoiceLanguage

The language of the voice (currently US English only).

Deprecated:

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Declared in NSSpeechSynthesizer.h.

NSVoiceLocaleIdentifier

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

NSVoiceSupportedCharacters

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

NSVoiceIndividuallySpokenCharacters

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

**Discussion**

Listing 1 lists the identifiers of the system voices (defined in /System/Library/Speech/Voices):

**Listing 1** Identifiers of the Mac OS X system voices

```
com.apple.speech.synthesis.voice.Agnes  
com.apple.speech.synthesis.voice.Albert  
com.apple.speech.synthesis.voice.BadNews  
com.apple.speech.synthesis.voice.Bahh  
com.apple.speech.synthesis.voice.Bells  
com.apple.speech.synthesis.voice.Boing  
com.apple.speech.synthesis.voice.Bruce  
com.apple.speech.synthesis.voice.Bubbles  
com.apple.speech.synthesis.voice.Cellos  
com.apple.speech.synthesis.voice.Deranged  
com.apple.speech.synthesis.voice.Fred  
com.apple.speech.synthesis.voice.GoodNews  
com.apple.speech.synthesis.voice.Hysterical  
com.apple.speech.synthesis.voice.Junior  
com.apple.speech.synthesis.voice.Kathy  
com.apple.speech.synthesis.voice.Organ  
com.apple.speech.synthesis.voice.Princess  
com.apple.speech.synthesis.voice.Ralph  
com.apple.speech.synthesis.voice.Trinoids
```

```
com.apple.speech.synthesis.voice.Vicki  
com.apple.speech.synthesis.voice.Victoria  
com.apple.speech.synthesis.voice.Whisper  
com.apple.speech.synthesis.voice.Zarvox
```

#### Availability

Available in Mac OS X v10.3 and later.

#### Declared In

NSSpeechSynthesizer.h

## Voice Genders

The `NSSpeechSynthesizer` class defines the following voice gender attributes, which are the allowable values of the `NSVoiceGender` key:

```
NSString *NSVoiceGenderNeuter;  
NSString *NSVoiceGenderMale;  
NSString *NSVoiceGenderFemale;
```

#### Constants

`NSVoiceGenderNeuter`

A neutral voice (neither male nor female)

Available in Mac OS X v10.3 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSVoiceGenderMale`

A male voice

Available in Mac OS X v10.3 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSVoiceGenderFemale`

A female voice

Available in Mac OS X v10.3 and later.

Declared in `NSSpeechSynthesizer.h`.

#### Availability

Available in Mac OS X v10.3 and later.

#### Declared In

NSSpeechSynthesizer.h

## Speech Synthesizer Property Keys

These constants identify synthesizer properties.

```

NSString *NSSpeechStatusProperty
NSString *NSSpeechErrorsProperty
NSString *NSSpeechInputModeProperty
NSString *NSSpeechCharacterModeProperty
NSString *NSSpeechNumberModeProperty
NSString *NSSpeechRateProperty
NSString *NSSpeechPitchBaseProperty
NSString *NSSpeechPitchModProperty
NSString *NSSpeechVolumeProperty
NSString *NSSpeechSynthesizerInfoProperty
NSString *NSSpeechRecentSyncProperty
NSString *NSSpeechPhonemeSymbolsProperty
NSString *NSSpeechCurrentVoiceProperty
NSString *NSSpeechCommandDelimiterProperty
NSString *NSSpeechResetProperty
NSString *NSSpeechOutputToFileURLProperty

```

**Declared In**

NSSpeechSynthesizer.h

## Input Modes

These constants identify input modes (used with NSSpeechInputModeProperty).

```

NSString *NSSpeechModeText
NSString *NSSpeechModePhoneme

```

**Declared In**

NSSpeechSynthesizer.h

## Speech Type

These constants identify speech types (used with NSSpeechInputModeProperty and NSSpeechNumberModeProperty).

```

NSString *NSSpeechModeNormal
NSString *NSSpeechModeLiteral

```

**Declared In**

NSSpeechSynthesizer.h

## Speech Status

These constants identify speech status (used with NSSpeechStatusProperty).

```

NSString *NSSpeechStatusOutputBusy
NSString *NSSpeechStatusOutputPaused
NSString *NSSpeechStatusNumberOfCharactersLeft
NSString *NSSpeechStatusPhonemeCode

```

**Declared In**

NSSpeechSynthesizer.h

## Synthesis Error

These constants identify errors that may occur during speech synthesis (used with `NSSpeechErrorProperty`).

```
NSString *NSSpeechErrorCount
NSString *NSSpeechErrorOldestCode
NSString *NSSpeechErrorOldestCharacterOffset
NSString *NSSpeechErrorNewestCode
NSString *NSSpeechErrorNewestCharacterOffset
```

### Declared In

`NSSpeechSynthesizer.h`

## Synthesizer Information

These constants identify synthesizer properties (used with `NSSpeechSynthesizerInfoProperty`).

```
NSString *NSSpeechSynthesizerInfoIdentifier
NSString *NSSpeechSynthesizerInfoVersion
```

### Declared In

`NSSpeechSynthesizer.h`

## Speech Command Delimiter

These constants identify speech-command delimiters in synthesized text (used with `NSSpeechCommandDelimiterProperty`).

```
NSString *NSSpeechCommandPrefix
NSString *NSSpeechCommandSuffix
```

### Declared In

`NSSpeechSynthesizer.h`

## Dictionary Properties

These constants identify speech-dictionary properties (use with `-addSpeechDictionary`).

```
NSString *NSSpeechDictionaryLanguage
NSString *NSSpeechDictionaryModificationDate
NSString *NSSpeechDictionaryPronunciations
NSString *NSSpeechDictionaryAbbreviations
NSString *NSSpeechDictionaryEntrySpelling
NSString *NSSpeechDictionaryEntryPhonemes
```

### Declared In

`NSSpeechSynthesizer.h`

## NSSpeechBoundary

These constants are used in stopping and pausing speech.

```
enum {
    NSSpeechImmediateBoundary = 0,
    NSSpeechWordBoundary,
    NSSpeechSentenceBoundary
};
typedef NSUInteger NSSpeechBoundary;
```

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

NSSpeechSynthesizer.h



# Document Revision History

---

This table describes the changes to *NSSpeechSynthesizer Class Reference*.

Date	Notes
2009-01-06	Clarified how the speech rate is set when using <code>initWithVoice:</code> .
2007-07-23	Updated for Mac OS X v10.5
2006-07-24	Clarified role and usage of the <code>NSVoiceIdentifier</code> constant. Added identifiers for the built-in voices.
2006-05-23	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History



# Index

---

## A

---

addSpeechDictionary: [instance method 12](#)  
attributesForVoice: [class method 11](#)  
availableVoices [class method 11](#)

## C

---

continueSpeaking [instance method 12](#)

## D

---

defaultVoice [class method 11](#)  
delegate [instance method 13](#)  
Dictionary Properties [28](#)

## I

---

initWithVoice: [instance method 13](#)  
Input Modes [27](#)  
isAnyApplicationSpeaking [class method 12](#)  
isSpeaking [instance method 14](#)

## N

---

NSSpeechBoundary [data type 28](#)  
NSVoiceAge [constant 24](#)  
NSVoiceDemoText [constant 25](#)  
NSVoiceGender [constant 25](#)  
NSVoiceGenderFemale [constant 26](#)  
NSVoiceGenderMale [constant 26](#)  
NSVoiceGenderNeuter [constant 26](#)  
NSVoiceIdentifier [constant 24](#)  
NSVoiceIndividuallySpokenCharacters [constant 25](#)

NSVoiceLanguage [constant \(Deprecated in Mac OS X v10.5\) 25](#)  
NSVoiceLocaleIdentifier [constant 25](#)  
NSVoiceName [constant 24](#)  
NSVoiceSupportedCharacters [constant 25](#)

## O

---

objectForProperty:error: [instance method 14](#)

## P

---

pauseSpeakingAtBoundary: [instance method 14](#)  
phonemesFromText: [instance method 15](#)

## R

---

rate [instance method 15](#)

## S

---

setDelegate: [instance method 16](#)  
setObject:forProperty:error: [instance method 16](#)  
setRate: [instance method 16](#)  
setUsesFeedbackWindow: [instance method 17](#)  
setVoice: [instance method 17](#)  
setVolume: [instance method 18](#)  
Speech Command Delimiter [28](#)  
Speech Status [27](#)  
Speech Synthesizer Property Keys [26](#)  
Speech Type [27](#)  
speechSynthesizer:didEncounterErrorAtIndex:  
ofString:message: <NSObject> [delegate method 21](#)  
speechSynthesizer:didEncounterSyncMessage:  
<NSObject> [delegate method 22](#)

speechSynthesizer:didFinishSpeaking:  
    <NSObject> delegate method [22](#)  
speechSynthesizer:willSpeakPhoneme: <NSObject>  
    delegate method [23](#)  
speechSynthesizer:willSpeakWord:ofString:  
    <NSObject> delegate method [23](#)  
startSpeakingString: instance method [18](#)  
startSpeakingString:toURL: instance method [19](#)  
stopSpeaking instance method [19](#)  
stopSpeakingAtBoundary: instance method [20](#)  
Synthesis Error [28](#)  
Synthesizer Information [28](#)

## U

---

usesFeedbackWindow instance method [20](#)

## V

---

Voice Attributes [24](#)  
Voice Genders [26](#)  
voice instance method [21](#)  
volume instance method [21](#)